A Swiss Army Infinitesimal Jackknife

Ryan Giordano
rgiordano@berkeley.edu

Will Stephenson
wtstephe@mit.edu

Runjing Liu
runjing_liu@berkeley.edu

Michael I. Jordan
jordan@cs.berkeley.edu

Tamara Broderick
tbroderick@csail.mit.edu

February 7, 2020

Abstract

The error or variability of machine learning algorithms is often assessed by repeatedly re-fitting a model with different weighted versions of the observed data. The ubiquitous tools of cross-validation (CV) and the bootstrap are examples of this technique. These methods are powerful in large part due to their model agnosticism but can be slow to run on modern, large data sets due to the need to repeatedly re-fit the model. In this work, we use a linear approximation to the dependence of the fitting procedure on the weights, producing results that can be faster than repeated re-fitting by an order of magnitude. This linear approximation is sometimes known as the "infinitesimal jackknife" in the statistics literature, where it is mostly used as a theoretical tool to prove asymptotic results. We provide explicit finite-sample error bounds for the infinitesimal jackknife in terms of a small number of simple, verifiable assumptions. Our results apply whether the weights and data are stochastic or deterministic, and so can be used as a tool for proving the accuracy of the infinitesimal jackknife on a wide variety of problems. As a corollary, we state mild regularity conditions under which our approximation consistently estimates true leave-k-out cross-validation for any fixed k. These theoretical results, together with modern automatic differentiation software, support the application of the infinitesimal jackknife to a wide variety of practical problems in machine learning, providing a "Swiss Army infinitesimal jackknife." We demonstrate the accuracy of our methods on a range of simulated and real datasets.

1 Introduction

Statistical machine learning methods are increasingly deployed in real-world problem domains where they are the basis of decisions affecting individuals'

employment, savings, health, and safety. Unavoidable randomness in data collection necessitates understanding how our estimates, and resulting decisions, might have differed had we observed different data. Both cross validation (CV) and the bootstrap attempt to diagnose this variation and are widely used in classical data analysis. But these methods are often prohibitively slow for modern, massive datasets, as they require running a learning algorithm on many slightly different datasets. In this work, we propose to replace these many runs with a single perturbative approximation. We show that the computation of this approximation is far cheaper than the classical methods, and we provide theoretical conditions that establish its accuracy.

Many data analyses proceed by minimizing a loss function of exchangeable data. Examples include empirical loss minimization and M-estimation based on product likelihoods. Since we typically do not know the true distribution generating the data, it is common to approximate the dependence of our estimator on the data via the dependence of the estimator on the empirical distribution. In particular, we often form a new, proxy dataset using random or deterministic modifications of the empirical distribution, such as randomly removing k datapoints for leave-k-out CV. A proxy dataset obtained in this way can be represented as a weighting of the original data. From a set of such proxy datasets we can obtain estimates of uncertainty, including estimates of bias, variance, and prediction accuracy.

As data and models grow, the cost of repeatedly solving a large optimization problem for a number of different values of weights can become impractically large. Conversely, though, larger datasets often exhibit greater regularity; in particular, under fairly general conditions, limit laws based on independence imply that an optimum exhibits diminishing dependence on any fixed set of data points. We use this observation to derive a linear approximation to resampling that needs to be calculated only once, but which nonetheless captures the variability inherent in the repeated computations of classical CV. Our method is an instance of the *infinitesimal jackknife* (IJ), a general methodology that was historically a precursor to cross-validation and the bootstrap [Jaeckel, 1972, Efron, 1982]. Part of our argument is that variants of the IJ should be reconsidered for modern large-scale applications because, for smooth optimization problems, the IJ can be calculated automatically with modern automatic differentiation tools [Baydin et al., 2017].

By using this linear approximation, we incur the cost of forming and inverting a matrix of second derivatives with size equal to the dimension of the parameter space, but we avoid the cost of repeatedly re-optimizing the objective. As we demonstrate empirically, this tradeoff can be extremely favorable in many problems of interest.

Our approach aims to provide a felicitous union of two schools of thought. In statistics, the IJ is typically used to prove normality or consistency of other estimators [Fernholz, 1983, Shao, 1993, Shao and Tu, 2012]. However, the conditions that are required for these asymptotic analyses to hold are prohibitively restrictive for machine learning—specifically, they require objectives with bounded gradients. A number of recent papers in machine learning have provided related

linear approximations for the special case of leave-one-out cross-validation [Koh and Liang, 2017, Rad and Maleki, 2018, Beirami et al., 2017], though their analyses lack the generality of the statistical perspective.

We combine these two approaches by modifying the proof of the Fréchet differentiability of M-estimators developed by Clarke [1983]. Specifically, we adapt the proof away from the question of Fréchet differentiability within the class of all empirical distributions to the narrower problem of approximating the exact re-weighting on a particular dataset with a potentially restricted set of weights. This limitation of what we expect from the approximation is crucial; it allows us to bound the error in terms of a complexity measure of the set of derivatives of the observed objective function, providing a basis for non-asymptotic applications in large-scale machine learning, even for objectives with unbounded derivatives. Together with modern automatic differentiation tools, these results extend the use of the IJ to a wider range of practical problems. Thus, our "Swiss Army infinitesimal jackknife," like the famous Swiss Army knife, is a single tool with many different functions.

2 Methods and Results

2.1 Problem definition

We consider the problem of estimating an unknown parameter $\theta \in \Omega_\theta \subseteq \mathbb{R}^D$, with a compact Ω_θ and a dataset of size N. Our analysis will proceed entirely in terms of a fixed dataset, though we will be careful to make assumptions that will plausibly hold for all N under suitably well-behaved random sampling. We define our estimate, $\hat{\theta} \in \Omega_\theta$, as the root of a weighted estimating equation. For each $n = 1, \ldots, N$, let $g_n(\theta)$ be a function from Ω_θ to $\mathbb{R}^D$. Let w_n be a real number, and let w be the vector collecting the w_n. Then $\hat{\theta}$ is defined as the quantity that satisfies

$$\hat{\theta}(w) := \quad \theta \text{ such that } \frac{1}{N}\sum_{n=1}^{N} w_n g_n(\theta) = 0. \tag{1}$$

We will impose assumptions below that imply at least local uniqueness of $\hat{\theta}(w)$; see the discussion following Assumption 2 in Section 2.3.

As an example, consider a family of continuously differentiable loss functions $f(\cdot, \theta)$ parameterized by θ and evaluated at data points $x_n, n = 1, \ldots, N$. If we want to solve the optimization problem $\hat{\theta} = \underset{\theta \in \Omega_\theta}{\operatorname{argmin}} \frac{1}{N}\sum_{n=1}^{N} f(x_n, \theta)$, then we take $g_n(\theta) = \partial f(x_n, \theta)/\partial\theta$ and $w_n \equiv 1$. By keeping our notation general, we will be able to analyze a more general class of problems, such as multi-stage optimization (see Section 6). However, to aid intuition, we will sometimes refer to the $g_n(\theta)$ as "gradients" and their derivatives as "Hessians."

When equation (1) is not degenerate (we articulate precise conditions below), $\hat{\theta}$ is a function of the weights through solving the estimating equation, and we

write $\hat{\theta}(w)$ to emphasize this. We will focus on the case where we have solved equation (1) for the weight vector of all ones, $1_w := (1, \ldots, 1)$, which we denote $\hat{\theta}_1 := \hat{\theta}(1_w)$.

A re-sampling scheme can be specified by choosing a set $W \subseteq \mathbb{R}^N$ of weight vectors. For example, to approximate leave-k-out CV, one repeatedly computes $\hat{\theta}(w)$ where w has k randomly chosen zeros and all ones otherwise. Define W_k as the set of every possible leave-k-out weight vector. Showing that our approximation is good for all leave-k-out analyses with probability one is equivalent to showing that the approximation is good for all $w \in W_k$.

In the case of the bootstrap, W contains a fixed number B of randomly chosen weight vectors, $w_b^* \overset{iid}{\sim} \text{Multinomial}\left(N, N^{-1}\right)$ for $b = 1, \ldots, B$, so that $\sum_{n=1}^N w_{bn}^* = N$ for each b. Note that while w_n or w_{bn}^* are scalars, w_b^* is a vector of length N. The distribution of $\hat{\theta}(w_b^*) - \hat{\theta}(1_w)$ is then used to estimate the sampling variation of $\hat{\theta}_1$. Define this set $W_B^* = \{w_1^*, \ldots, w_B^*\}$. Note that W_B^* is stochastic and is a subset of all weight vectors that sum to N.

In general, W can be deterministic or stochastic, may contain integer or non-integer values, and may be determined independently of the data or jointly with it. As with the data, our results hold for a given W, but in a way that will allow natural high-probability extensions to stochastic W.

2.2 Linear approximation

The main problem we solve is the computational expense involved in evaluating $\hat{\theta}(w)$ for all the $w \in W$. Our contribution is to use only quantities calculated from $\hat{\theta}_1$ to approximate $\hat{\theta}(w)$ for all $w \in W$, without re-solving equation (1). Our approximation is based on the derivative $\frac{d\hat{\theta}(w)}{dw^T}$, whose existence depends on the derivatives of $g_n(\theta)$, which we assume to exist, and which we denote as $h_n(\theta) := \frac{\partial g_n(\theta)}{\partial \theta^T}$. We use this notation because $h_n(\theta)$ would be the Hessian of a term of the objective in the case of an optimization problem. We make the following definition for brevity.

Definition 1. The fixed point equation and its derivative are given respectively by

$$G(\theta, w) := \frac{1}{N} \sum_{n=1}^N w_n g_n(\theta)$$
$$H(\theta, w) := \frac{1}{N} \sum_{n=1}^N w_n h_n(\theta).$$

Note that $G\left(\hat{\theta}(w), w\right) = 0$ because $\hat{\theta}(w)$ solves equation (1) for w. We define $H_1 := H\left(\hat{\theta}_1, 1_w\right)$ and define the weight difference as $\Delta w = w - 1_w \in \mathbb{R}^N$. When H_1 is invertible, one can use the implicit function theorem and the chain rule to

show that the derivative of $\hat{\theta}(w)$ with respect to w is given by

$$\begin{aligned}\frac{d\hat{\theta}(w)}{dw^T}|_{1_w}\Delta w &= -H_1^{-1}\frac{1}{N}\sum_{n=1}^{N} g_n\left(\hat{\theta}_1\right)\Delta w \\ &= -H_1^{-1}G\left(\hat{\theta}_1, \Delta w\right).\end{aligned}$$

This derivative allows us to form a first-order approximation to $\hat{\theta}(w)$ at $\hat{\theta}_1$.

Definition 2. Our linear approximation to $\hat{\theta}(w)$ is given by

$$\hat{\theta}_{\mathrm{IJ}}(w) := \hat{\theta}_1 - H_1^{-1}G\left(\hat{\theta}_1, \Delta w\right).$$

We use the subscript "IJ" for "infinitesimal jackknife," which is the name for this estimate in the statistics literature [Jaeckel, 1972, Shao, 1993]. Because $\hat{\theta}_{\mathrm{IJ}}$ depends only on $\hat{\theta}_1$ and Δw, and not on solutions at any other values of w, there is no need to re-solve equation (1). Instead, to calculate $\hat{\theta}_{\mathrm{IJ}}$ one must solve a linear system involving H_1. Recalling that θ is D-dimensional, the calculation of H_1^{-1} (or a factorization that supports efficient solution of linear systems) can be $O\left(D^3\right)$. However, once H_1^{-1} is calculated or H_1 is factorized, calculating our approximation $\hat{\theta}_{\mathrm{IJ}}(w)$ for each new weight costs only as much as a single matrix-vector multiplication. Furthermore, H_1 often has a sparse structure allowing H_1^{-1} to be calculated more efficiently than a worst-case scenario (see Section 6 for an example). In more high-dimensional examples with dense Hessian matrices, such as neural networks, one may need to turn to approximations such as stochastic second-order methods [Koh and Liang, 2017, Agarwal et al., 2017] and conjugate gradient [Wright and Nocedal, 1999]. Indeed, even in relatively small or sparse problems, the vast bulk of the computation required to calculate $\hat{\theta}_{\mathrm{IJ}}$ is in the computation of H_1^{-1}. We leave the important question of approximate calculation of H_1^{-1} for future work.

2.3 Assumptions and results

We now state our key assumptions and results, which are sufficient conditions under which $\hat{\theta}_{\mathrm{IJ}}(w)$ will be a good approximation to $\hat{\theta}(w)$. We defer most proofs to Appendix A. We use $\|\cdot\|_{op}$ to denote the matrix operator norm, $\|\cdot\|_2$ to denote the L_2 norm, and $\|\cdot\|_1$ to denote the L_1 norm. For quantities like g and h, which have dimensions $N \times D$ and $N \times D \times D$ respectively, we apply the L_p norm to the vectorized version of arrays. For example, $\frac{1}{\sqrt{N}}\|h(\theta)\|_2 = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{D}\sum_{j=1}^{D}\left[h_n(\theta)\right]_{ij}^2}$ which is the square root of a sample average over $n \in [N]$.

We state all assumptions and results for a fixed N, a given estimating equation vector $g(\theta)$, and a fixed class of weights W. Although our analysis proceeds with these quantities fixed, we are careful to make only assumptions that can plausibly hold for all N and/or for randomly chosen W under appropriate regularity conditions.

Assumption 1 (Smoothness)**.** *For all $\theta \in \Omega_\theta$, each $g_n(\theta)$ is continuously differentiable in θ.*

The smoothness in Assumption 1 is necessary for a local approximation like Definition 2 to have any hope of being useful.

Assumption 2 (Non-degeneracy)**.** *For all $\theta \in \Omega_\theta$, $H(\theta, 1_w)$ is non-singular, with $\sup_{\theta\in\Omega_\theta} \left\| H(\theta, 1_w)^{-1} \right\|_{op} \leq C_{op} < \infty$.*

Without Assumption 2, the derivative in Definition 2 would not exist. For an optimization problem, Definition 2 amounts to assuming that the Hessian is strongly positive definite, and, in general, assures that the solution $\hat{\theta}_1$ is unique. Under our assumptions, we will show later that, additionally, $\hat{\theta}(w)$ is unique in a neighborhood of $\hat{\theta}_1$; see Lemma 6 of Appendix A. Furthermore, by fixing C_{op}, if we want to apply Assumption 2 for $N \to \infty$, we will require that H_1 remains strongly positive definite.

Assumption 3 (Bounded averages)**.** *There exist finite constants C_g and C_h such that $\sup_{\theta\in\Omega_\theta} \frac{1}{\sqrt{N}} \|g(\theta)\|_2 \leq C_g < \infty$ and $\sup_{\theta\in\Omega_\theta} \frac{1}{\sqrt{N}} \|h(\theta)\|_2 \leq C_h < \infty$.*

Assumption 3 essentially states that the sample variances of the gradients and Hessians are uniformly bounded. Note that it does not require that these quantities are bounded term-wise. For example, we allow $\sup_n \|g_n(\theta)\|_2^2 \underset{N\to\infty}{\longrightarrow} \infty$, as long as $\sup_n \frac{1}{N} \|g_n(\theta)\|_2^2$ remains bounded. This is a key advantage of the present work over many past applications of the IJ to M-estimation, which require $\sup_n \|g_n(\theta)\|_2^2$ to be uniformly bounded for all N [Shao and Tu, 2012, Beirami et al., 2017].

In both machine learning and statistics, $\sup_n \|g_n(\theta)\|_2^2$ is rarely bounded, though $\frac{1}{N} \|g(\theta)\|_2^2$ often is. As a simple example, suppose that $\theta \in \mathbb{R}^1$, $x_n \sim \mathcal{N}(0,1)$, and $g_n = \theta - x_n$, as would arise from the squared error loss $f_n(x_n, \theta) = \frac{1}{2}(\theta - x_n)^2$. Fix a θ and let $N \to \infty$. Then $\sup_n \|g_n(\theta)\|_2^2 \to \infty$ because $\sup_n |x_n| \to \infty$, but $\frac{1}{N} \|g(\theta)\|_2^2 \to \theta^2 + 1$ by the law of large numbers.

Assumption 4 (Local smoothness)**.** *There exists a $\Delta_\theta > 0$ and a finite constant L_h such that, $\left\| \theta - \hat{\theta}_1 \right\|_2 \leq \Delta_\theta$ implies that $\frac{\left\| h(\theta) - h(\hat{\theta}_1) \right\|_2}{\sqrt{N}} \leq L_h \left\| \theta - \hat{\theta}_1 \right\|_2$.*

The constants defined in Assumption 4 are needed to calculate our error bounds explicitly.

Assumptions 1–4 are quite general and should be expected to hold for many reasonable problems, including holding uniformly asymptotically with high probability for many reasonable data-generating distributions, as the following lemma shows.

Lemma 1 (The assumptions hold under uniform convergence)**.** *Let Ω_θ be a compact set, and let $g_n(\theta)$ be twice continuously differentiable IID random functions for $n \in [N]$. (The function is random but θ is not—for example,*

$\mathbb{E}[g_n(\theta)]$ *is still a function of* θ*.) Define* $r_n(\theta) := \frac{\partial^2 g_n(\theta)}{\partial\theta\partial\theta}$*, so* $r_n(\theta)$ *is a* $D \times D \times D$ *tensor.*

Assume that we can exchange integration and differentiation, that $\mathbb{E}[h_n(\theta)]$ *is non-singular for all* $\theta \in \Omega_\theta$*, and that all of* $\mathbb{E}\left[\sup_{\theta\in\Omega_\theta} \|g_n(\theta)\|_2^2\right]$, $\mathbb{E}\left[\sup_{\theta\in\Omega_\theta} \|h_n(\theta)\|_2^2\right]$, *and* $\mathbb{E}\left[\sup_{\theta\in\Omega_\theta} \|r_n(\theta)\|_2^2\right]$ *are finite.*

Then $\lim_{N\to\infty} P(\textit{Assumptions 1–4 hold}) = 1$.

Lemma 1 follows from the uniform convergence results of Theorems 9.1 and 9.2 in Keener [2011]. See Appendix A.4 for a detailed proof. A common example to which Lemma 1 would apply is where x_n are well-behaved IID data and $g_n(\theta) = \gamma(x_n, \theta)$ for an appropriately smooth estimating function $\gamma(\cdot, \theta)$. See Keener [2011, Chapter 9] for more details and examples, including applications to maximum likelihood estimators on unbounded domains.

Assumptions 1–4 apply to the estimating equation. We also require a boundedness condition for W.

Assumption 5 (Bounded weight averages)**.** *The quantity* $\frac{1}{\sqrt{N}}\|w\|_2$ *is uniformly bounded for* $w \in W$ *by a finite constant* C_w.

Our final requirement is considerably more restrictive, and contains the essence of whether or not $\hat{\theta}_{\mathrm{IJ}}(w)$ will be a good approximation to $\hat{\theta}(w)$.

Condition 1 (Set complexity)**.** *There exists a* $\delta \geq 0$ *and a corresponding set* $W_\delta \subseteq W$ *such that*

$$\max_{w\in W_\delta} \sup_{\theta\in\Omega_\theta} \left\| \frac{1}{N}\sum_{n=1}^N (w_n - 1)\, g_n(\theta) \right\|_1 \leq \delta \quad \textit{and}$$
$$\max_{w\in W_\delta} \sup_{\theta\in\Omega_\theta} \left\| \frac{1}{N}\sum_{n=1}^N (w_n - 1)\, h_n(\theta) \right\|_1 \leq \delta.$$

Condition 1 is central to establishing when the approximation $\hat{\theta}_{\mathrm{IJ}}(w)$ is accurate. For a given δ, W_δ will be the class of weight vectors for which $\hat{\theta}_{\mathrm{IJ}}(w)$ is accurate to within order δ. Trivially, $1_w \in W_\delta$ for $\delta = 0$, so W_δ is always non-empty, even for arbitrarily small δ. The trick will be to choose a small δ that still admits a large class W_δ of weight vectors. In Section 3 we will discuss Condition 1 in more depth, but it will help to first state our main theorem.

Definition 3. The following constants are given by quantities in Assumptions 1–5 .

$$C_{\mathrm{IJ}} := 1 + DC_w L_h C_{op}$$
$$\Delta_\delta := \min\left\{ \Delta_\theta C_{op}^{-1}, \frac{1}{2} C_{\mathrm{IJ}}^{-1} C_{op}^{-1} \right\}.$$

Note that, although the parameter dimension D occurs explicitly only once in Definition 3, all of C_w, C_{op}, and L_h in general might also contain dimension

dependence. Additionally, the bound δ in Condition 1, a measure of the set complexity of the parameters, will typically depend on dimension. However, the particular place where the parameter dimension enters will depend on the problem and asymptotic regime, and our goal is to provide an adaptable toolkit for a wide variety of problems.

We are now ready to state our main result.

Theorem 1 (Error bound for the approximation)**.** *Under Assumptions 1–5 and Condition 1,*

$$\delta \leq \Delta_\delta \Rightarrow \max_{w \in W_\delta} \left\| \hat{\theta}_{IJ}(w) - \hat{\theta}(w) \right\|_2 \leq 2C_{op}^2 C_{IJ} \delta^2.$$

We stress that Theorem 1 bounds only the difference between $\hat{\theta}_{\mathrm{IJ}}(w)$ and $\hat{\theta}(w)$. Theorem 1 alone does not guarantee that $\hat{\theta}_{\mathrm{IJ}}(w)$ converges to any hypothetical infinite population quantity. We see this as a strength, not a weakness. To begin with, convergence to an infinite population requires stronger assumptions. Contrast, for example, the Fréchet differentiability work of Clarke [1983], on which our work is based, with the stricter requirements in the proof of consistency in Shao [1993]. Second, machine learning problems may not naturally admit a well-defined infinite population, and the dataset at hand may be of primary interest. Finally, by analyzing a particular sample rather than a hypothetical infinite population, we can bound the error in terms of the quantities C_{IJ} and Δ_δ, which can actually be calculated from the data at hand.

Still, Theorem 1 is useful to prove asymptotic results about the difference $\left\| \hat{\theta}_{\mathrm{IJ}}(w) - \hat{\theta}(w) \right\|_2$. As an illustration, we now show that the uniform consistency of leave-k-out CV follows from Theorem 1 by a straightforward application of Hölder's inequality.

Corollary 1 (Consistency for leave-k-out CV)**.** *Assume that Assumptions 1–5 hold uniformly for all N. Fix an integer k, and let*

$$W_k := \{w : w_n = 0 \text{ in } k \text{ entries and } 1 \text{ otherwise}\}.$$

Then, for all N, there exists a constant C_K such that

$$\begin{aligned} \sup_{w \in W_k} \left\| \hat{\theta}_{IJ}(w) - \hat{\theta}(w) \right\|_2 &\leq C_K \frac{\|g\|_\infty^2}{N^2} \\ &\leq C_K \frac{\max\{C_g, C_h\}^2}{N}. \end{aligned}$$

Proof. For $w \in W_k$, $\frac{\|\Delta w\|_2}{\sqrt{N}} = \sqrt{\frac{K}{N}}$. Define $C_{gh} := \max\{C_g, C_h\}$. By Assumption 3, $\|g\|_2 / \sqrt{N} \leq C_{gh}$ and $\|h\|_2 / \sqrt{N} \leq C_{gh}$ for all N. By Hölder's inequality,

$$\begin{aligned} &\sup_{w \in W} \sup_{\theta \in \Omega_\theta} \left\| \frac{1}{N} \sum_{n=1}^{N} (w_n - 1) g_n(\theta) \right\|_1 \\ &\leq \sup_{w \in W} \|w - 1_w\|_1 \sup_{\theta \in \Omega_\theta} \frac{\|g\|_\infty}{N} = K \frac{\|g\|_\infty}{N} \leq K \frac{C_{gh}}{\sqrt{N}}, \end{aligned}$$

with a similar bound for $\|h\|_2$. Consequently, for N large enough, Condition 1 is satisfied with $W_\delta = W_k$ and either $\delta = K\frac{\|g\|_\infty}{N}$ or $\delta = K\frac{C_{gh}}{\sqrt{N}}$. The result then follows from Theorem 1. □

3 Examples

The moral of Theorem 1 is that, under Assumptions 1–5 and Condition 1, $\left\|\hat{\theta}_{\text{IJ}} - \hat{\theta}(w)\right\| = O\left(\delta^2\right)$ for $w \in W_\delta$. That is, if we can make δ small enough, W_δ big enough, and still satisfy Condition 1, then $\hat{\theta}_{\text{IJ}}(w)$ is a good approximation to $\hat{\theta}(w)$ for "most" w, where "most" is defined as the size of W_δ. So it is worth taking a moment to develop some intuition for Condition 1. We have already seen in Corollary 1 that $\hat{\theta}_{\text{IJ}}$ is, asymptotically, a good approximation for leave-k-out CV uniformly in W. We now discuss some additional cases: first, a worst-case example for which $\hat{\theta}_{\text{IJ}}$ is not expected to work, second the bootstrap, and finally we revisit leave-one-out cross validation in the context of these other two methods.

First, consider a pathological example. Let W_{full} be the set of all weight vectors that sum to N. Let $n^* = \max_{n\in[N]} \left\|g_n\left(\hat{\theta}_1\right)\right\|_1$ be the index of the gradient term with the largest L_1 norm, and let $w_{n^*} = N$ and $w_n = 0$ for $n \neq n^*$. Then

$$\begin{aligned}&\sup_{\theta\in\Omega_\theta}\left\|\frac{1}{N}\sum_{n=1}^{N}(w_n - 1)\,g_n(\theta)\right\|_1 \\ &\quad= \sup_{\theta\in\Omega_\theta}\left\|g_{n^*}(\theta) - \frac{1}{N}\sum_{n=1}^{N} g_n(\theta)\right\|_1 \geq \left\|g_{n^*}\left(\hat{\theta}_1\right)\right\|_1.\end{aligned}$$

(The last inequality uses the fact that $G\left(\hat{\theta}_1, 1_w\right) = 0$.) In this case, unless the largest gradient, $\left\|g_{n^*}\left(\hat{\theta}_1\right)\right\|_1$, is small, Condition 1 will not be satisfied for small δ, and we would not expect $\hat{\theta}_{\text{IJ}}$ to be a good estimate for $\hat{\theta}(w)$ for all $w \in W_{full}$. The class W_{full} is too expressive. In the language of Condition 1, for some small fixed δ, W_δ will be some very restricted subset of W_{full} in most realistic situations.

Now, suppose that we are using B bootstrap weights, $w_b^* \overset{iid}{\sim} \text{Multinomial}\left(N, N^{-1}\right)$ for $b = 1, ..., B$, and analyzing an optimization problem as defined in Section 2.1. For a given w_b^*, a dataset $x_1^*, ..., x_N^*$ formed by taking $w_{b,n}^*$ copies of datapoint x_n is equivalent in distribution to N IID samples with replacement from the

empirical distribution on $(x_1, ..., x_N)$. In this notation, we then have

$$\frac{1}{N}\sum_{n=1}^{N}\left(w_b^* - 1\right) g_n\left(\theta\right) = \\ \frac{1}{N}\sum_{n=1}^{N}\frac{\partial f\left(\theta, x_n^*\right)}{\partial\theta} - \frac{1}{N}\sum_{n=1}^{N}\frac{\partial f\left(\theta, x_n\right)}{\partial\theta}.$$

In this case, Condition 1 is a uniform bound on a centered empirical process of derivatives of the objective function. Note that estimating sample variances by applying the IJ with bootstrap weights is equivalent to the ordinary delta method based on an asymptotic normal approximation [Efron, 1982, Chapter 21]. In order to provide an approximation to the bootstrap that retains benefits (such as the faster-than-normal convergence to the true sampling distribution described by Hall [2013]), one must consider higher-ordered Taylor expansions of $\hat{\theta}(w)$. We leave this for future work.

Finally, let us return to leave-one-out CV. In this case, $w_n - 1$ is nonzero for exactly one entry. Again, we can choose to leave out the adversarially-chosen n^* as in the first pathological example. However, unlike the pathological example, the leave-one-out CV weights are constrained to be closer to 1_w—specifically, we set $w_{n^*} = 0$, and let w be one elsewhere. Then Condition 1 requires $\sup_{\theta\in\Omega_\theta}\left\|\frac{1}{N}g_{n^*}\left(\theta\right)\right\|_1 \leq \delta$. In contrast to the pathological example, this supremum will get smaller as N increases as long as $\left\|g_{n^*}\left(\theta\right)\right\|_1$ grows more slowly than N. For this reason, we expect leave-one-out (and, indeed, leave-k-out for fixed k) to be accurately approximated by $\hat{\theta}_{\text{IJ}}$ in many cases of interest, as stated in Corollary 1.

4 Related Work

Although the idea of forming a linear approximation to the re-weighting of an M-estimator has a long history, we nevertheless contribute in a number of ways. By limiting ourselves to approximating the exact reweighting on a particular dataset, we both loosen the strict requirements from the statistical literature and generalize the existing results from the machine learning literature.

The jackknife is often favored over the IJ in the statistics literature because of the former's simple computational approach, as well as perceived difficulties in calculating the necessary derivatives when some of the parameters are implicitly defined via optimization [Shao and Tu, 2012, Chapter 2.1] (though exceptions exist; see, e.g., Wager et al. [2014]). The brute-force approach of the jackknife is, however, a liability in large-scale machine learning problems, which are generally extremely expensive to re-optimize. Furthermore, and critically, the complexity and tedium of calculating the necessary derivatives is entirely eliminated by modern automatic differentiation [Baydin et al., 2017, Maclaurin et al., 2015].

Our work is based on the proof of the Fréchet differentiability of M-estimators of Clarke [1983]. In classical statistics, Fréchet differentiability is typically used

to describe the asymptotic behavior of functionals of the empirical distribution in terms of a functional [Mises, 1947, Fernholz, 1983]. Since Clarke [1983] was motivated by such asymptotic questions, he studied the Fréchet derivative evaluated at a continuous probability distribution for function classes that included delta functions. This focus led to the requirement of a bounded gradient. However, unbounded gradients are ubiquitous in both statistics and machine learning, and an essential contribution of the current paper is to remove the need for bounded gradients.

There exist proofs of the consistency of the (non-infinitesimal) jackknife that allow for unbounded gradients. For example, it is possible that the proofs of Reeds [1978], which require a smoothness assumption similar to our Assumption 4, could be adapted to the IJ. However, the results of Reeds [1978]—as well as those of Clarke [1983] and subsequent applications such as those of Shao and Tu [2012]—are asymptotic and applicable only to IID data. By providing finite sample results for a fixed dataset and weight set, we are able to provide a template for proving accuracy bounds for more generic probability distributions and re-weighting schemes.

A number of recent machine learning papers have derived approximate linear versions of leave-one-out estimators. Koh and Liang [2017] consider approximating the effect of leaving out one observation at a time to discover influential observations and construct adversarial examples, but provide little supporting theory. Beirami et al. [2017] provide rigorous proofs for an approximate leave-one-out CV estimator; however, their estimator requires computing a new inverse Hessian for each new weight at the cost of a considerable increase in computational complexity. Like the classical statistics literature, Beirami et al. [2017] assume that the gradients are bounded for all N. When $\|g\|_\infty^2$ in Corollary 1 is finite for all N, we achieve the same N^{-2} rate claimed by Beirami et al. [2017] for leave-one-out CV although we use only a single matrix inverse. Rad and Maleki [2018] also approximate leave-one-out CV, and prove tighter bounds for the error of their approximation than we do, but their work is customized to leave-one-out CV and makes much more restrictive assumptions (e.g., Gaussianity).

5 Simulated Experiments

We begin the empirical demonstration of our method on two simple generalized linear models: logistic and Poisson regression.[1] In each case, we generate a synthetic dataset $Z = \{(x_n, y_n)\}_{n=1}^N$ from parameters (θ, b), where $\theta \in \mathbb{R}^{100}$ is a vector of regression coefficients and $b \in \mathbb{R}$ is a bias term. In each experiment, $x_n \in \mathbb{R}^{100}$ is drawn from a multivariate Gaussian, and y_n is a scalar drawn from a Bernoulli distribution with the logit link or from a Poisson distribution with the exponential link.

[1] Leave-one-out CV may not be the most appropriate estimator of generalization error in this setting [Rosset and Tibshirani, 2018], but this section is intended only to provide simple illustrative examples.

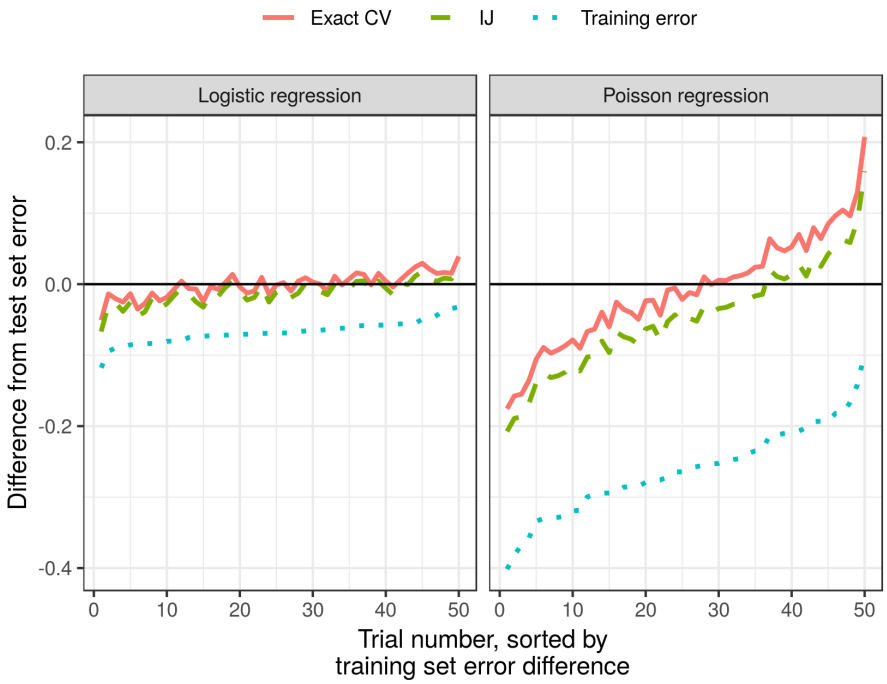

Figure 1: Simulated data: accuracy results.

For a ground truth, we generate a large test set with $N = 100{,}000$ datapoints to measure the true generalization error. We show in Fig. 1 that, over 50 randomly generated datasets, our approximation consistently underestimates the actual error predicted by exact leave-one-out CV; however, the difference is small relative to the improvements they both make over the error evaluated on the training set.

Fig. 2 shows the relative timings of our approximation and exact leave-one-out CV on logistic regression with datasets of increasing size. The time to run our approximation is roughly an order of magnitude smaller.

6 Genomics Experiments

We now consider a genomics application in which we use CV to choose the degree of a spline smoother when clustering time series of gene expression data. Code and instructions to reproduce our results can be found in the git repository rgiordan/AISTATS2019SwissArmyIJ. The application is also described in detail in Appendix B.

We use a publicly available data set of mice gene expression [Shoemaker et al., 2015] in which mice were infected with influenza virus, and gene expression was

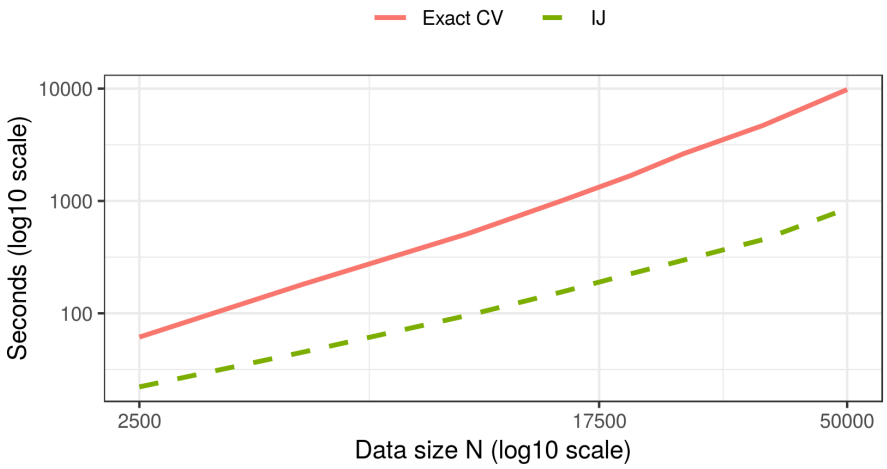

Figure 2: Simulated data: timing results.

assessed several times after infection. The observed data consists of expression levels y_{gt} for genes $g = 1, \ldots, n_g$ and time points $t = 1, \ldots, n_t$. In our case $n_g = 1000$ and $n_t = 14$. Many genes behave the same way; thus, clustering the genes by the pattern of their behavior over time allows dimensionality reduction that can facilitate interpretation. Consequently, we wish to first fit a smoothed regression line to each gene and then cluster the results. Following Luan and Li [2003], we model the time series as a gene-specific constant additive offset plus a B-spline basis of degree 3, and the task is to choose the B-spline basis degrees of freedom using cross-validation on the time points.

Our analysis runs in two stages—first, we regress the genes on the spline basis, and then we cluster a transformed version of the regression fits. By modeling in two stages, we both speed up the clustering and allow for the use of flexible transforms of the fits. We are interested in choosing the smoothing parameter using CV on the time points. Both the time points and the smoothing parameter enter the regression objective directly, but they affect the clustering objective only through the optimal regression parameters. Because the optimization proceeds in two stages, the fit is not the optimum of any single objective function. However, it can still be represented as an M-estimator (see Appendix B).

We implemented the model in `scipy` [Jones et al., 2001] and computed all derivatives with `autograd` [Maclaurin et al., 2015]. We note that the match between "exact" cross-validation (removing time points and re-optimizing) and the IJ was considerably improved by using a high-quality second-order optimization method. In particular, for these experiments, we employed the Newton conjugate-gradient trust region method [Wright and Nocedal, 1999, Chapter 7.1] as implemented by the method `trust-ncg` in `scipy.optimize`, preconditioned by the Cholesky decomposition of an inverse Hessian calculated at an initial

approximate optimum. The Hessian used for the preconditioner was with respect to the clustering parameters only and so could be calculated quickly, in contrast to the H_1 matrix used for the IJ, which includes the regression parameters as well. We found that first-order or quasi-Newton methods (such as BFGS) often got stuck or terminated at points with fairly large gradients. At such points our method does not apply in theory nor, we found, very well in practice.

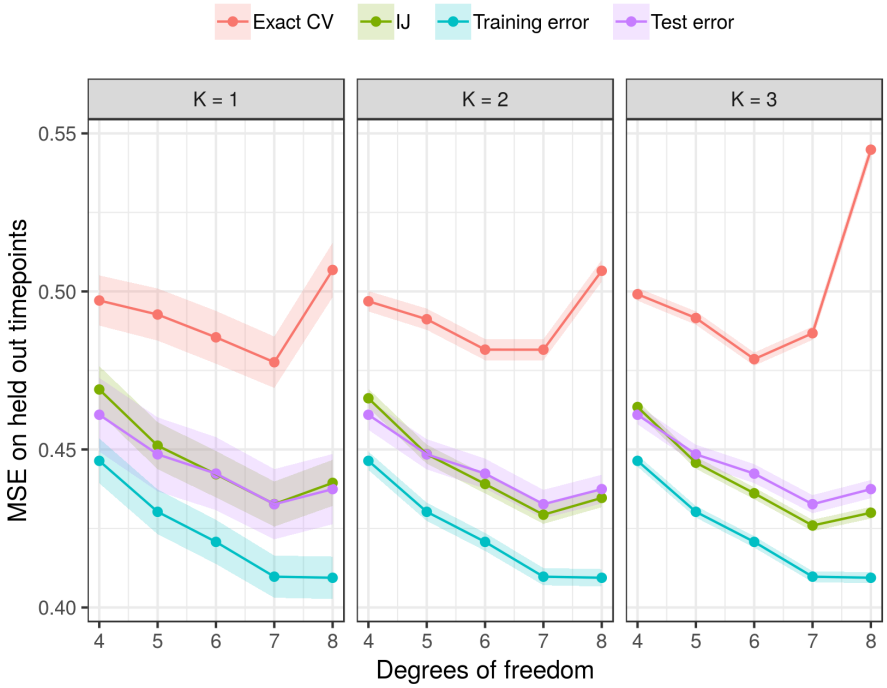

Figure 3: Genomics data: accuracy results.

Fig. 3 shows that the IJ is a reasonably good approximation to the test set error.[2] In particular, both the IJ and exact CV capture the increase in test error for $df = 8$, which is not present in the training error. Thus we see that, like exact CV, the IJ is able to prevent overfitting. Though the IJ underestimates exact CV, we note that it differs from exact CV by no more than exact CV itself differs from the true quantity of iterest, the test error.

The timing results for the genomics experiment are shown in Fig. 4. For this particular problem with approximately $39,000$ parameters (the precise number depends on the degrees of freedom), finding the initial optimum takes about 42 seconds. The cost of finding the initial optimum is shared by exact CV and the

[2]In fact, in this case, the IJ is a better predictor of test set error than exact CV. However, the authors have no reason at present to believe that the IJ is a better predictor of test error than exact CV in general.

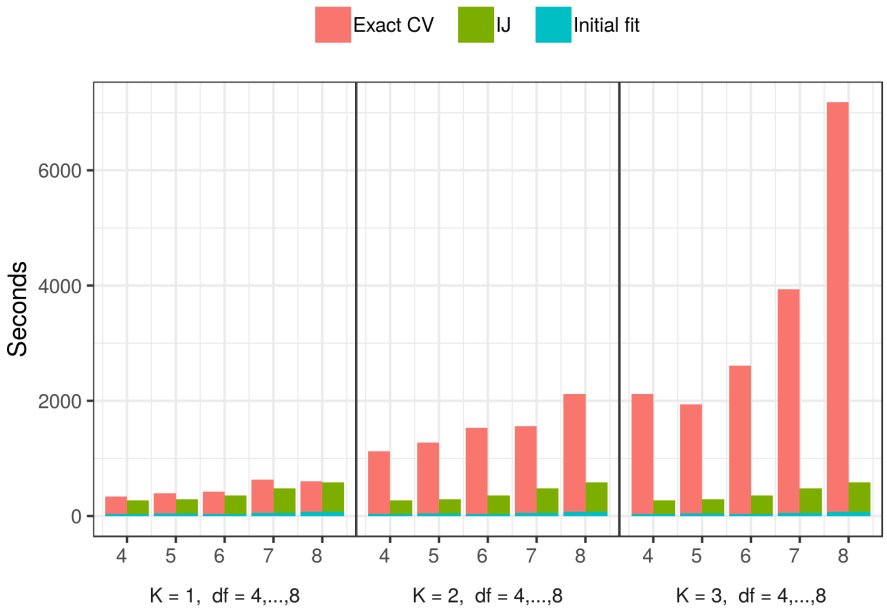

Figure 4: Genomics data: timing results.

IJ, and, as shown in Fig. 4, is a small proportion of both.

The principle time cost of the IJ is the computation of H_1. Computing and inverting a dense matrix of size $39,000$ would be computationally prohibitive. But, for the regression objective, H_1 is extremely sparse and block diagonal, so computing H_1 in this case took only around 360 seconds. Inverting H_1 took negligible time. Once we have H_1^{-1}, obtaining the subsequent IJ approximations is nearly instantaneous.

The cost of refitting the model for exact CV varies by degrees of freedom (increasing degrees of freedom increases the number of parameters) and the number of left-out points (an increasing number of left-out datapoints increases the number of refits). As can be seen in Fig. 4, for low degrees of freedom and few left-out points, the cost of re-optimizing is approximately the same as the cost of computing H_1. However, as the degrees of freedom and number of left-out points grow, the cost of exact CV increases to as much as an order of magnitude more than that of the IJ.

7 Conclusion

We recommend consideration of the Swiss Army infinitesimal jackknife for modern machine learning problems. The large size of modern data both increases the need for fast approximations and renders such approximations more accurate.

Furthermore, modern automatic differentiation renders many past practical difficulties obsolete. By stepping back from the strict requirements of classical statistical theory, we can see that the value of the infinitesimal jackknife extends beyond its traditional application areas, while retaining desirable generality in other respects.

Acknowledgements. We thank anonymous reviewers for their helpful comments and suggestions. We are grateful to Nelle Varoquaux for her help with the genomics experiments and to Pang Wei Koh for pointing out and helping to correct an error in an earlier version of our proofs. This research was supported in part by DARPA (FA8650-18-2-7832), an ARO YIP award, an NSF CAREER award, and the CSAIL-MSR Trustworthy AI Initiative. Ryan Giordano was supported by the Gordon and Betty Moore Foundation through Grant GBMF3834 and by the Alfred P. Sloan Foundation through Grant 2013-10-27 to the University of California, Berkeley. Runjing Liu was supported by the NSF Graduate Research Fellowship.

References

N. Agarwal, B. Bullins, and E. Hazan. Second-order stochastic optimization in linear time. *Journal of Machine Learning Research*, 2017.

A. Baydin, B. Pearlmutter, A. Radul, and J. Siskind. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18(153): 1–153, 2017.

A. Beirami, M. Razaviyayn, S. Shahrampour, and V. Tarokh. On optimal generalizability in parametric learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3458–3468, 2017.

B. Clarke. Uniqueness and Fréchet differentiability of functional solutions to maximum likelihood type equations. *The Annals of Statistics*, 11(4):1196–1205, 1983.

R. Dudley. *Real analysis and probability*. Chapman and Hall/CRC, 2018.

B. Efron. *The Jackknife, the Bootstrap, and Other Resampling Plans*, volume 38. Society for Industrial and Applied Mathematics, 1982.

L. Fernholz. *Von Mises Calculus for Statistical Functionals*, volume 19. Springer Science & Business Media, 1983.

P. Hall. *The Bootstrap and Edgeworth Expansion*. Springer Science & Business Media, 2013.

L. Jaeckel. The infinitesimal jackknife, memorandum. Technical report, MM 72-1215-11, Bell Lab. Murray Hill, NJ, 1972.

E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001. URL `http://www.scipy.org/`.

R. W. Keener. *Theoretical Statistics: Topics for a Core Course*. Springer, 2011.

P. W. Koh and P. Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning (ICML)*, 2017.

Y. Luan and H. Li. Clustering of time-course gene expression data using a mixed-effects model with B-splines. *Bioinformatics*, 19(4):474–482, 2003.

D. Maclaurin, D. Duvenaud, and R. P. Adams. Autograd: Effortless gradients in numpy. In *International Conference on Machine Learning (ICML) AutoML Workshop*, 2015.

R. Mises. On the asymptotic distribution of differentiable statistical functions. *The Annals of Mathematical Statistics*, 18(3):309–348, 1947.

K. R. Rad and A. Maleki. A scalable estimate of the extra-sample prediction error via approximate leave-one-out. *arXiv Preprint*, January 2018.

J. A. Reeds. Jackknifing maximum likelihood estimates. *The Annals of Statistics*, pages 727–739, 1978.

S. Rosset and R. J. Tibshirani. From fixed-X to random-X regression: Bias-variance decompositions, covariance penalties, and prediction error estimation. *Journal of the American Statistical Association*, 2018.

J. Schott. *Matrix Analysis for Statistics*. John Wiley & Sons, 2016.

J. Shao. Differentiability of statistical functionals and consistency of the jackknife. *The Annals of Statistics*, pages 61–75, 1993.

J. Shao and D. Tu. *The Jackknife and Bootstrap*. Springer Series in Statistics, 2012.

J. E. Shoemaker, S. Fukuyama, A. J. Eisfeld, D. Zhao, E. Kawakami, S. Sakabe, T. Maemura, T. Gorai, H. Katsura, Y. Muramoto, S. Watanabe, T. Watanabe, K. Fuji, Y. Matsuoka, H. Kitano, and Y. Kawaoka. An ultrasensitive mechanism regulates influenza virus-induced inflammation. *PLoS Pathogens*, 11(6):1–25, 2015.

S. Wager, T. Hastie, and B. Efron. Confidence intervals for random forests: The jackknife and the infinitesimal jackknife. *The Journal of Machine Learning Research*, 15(1):1625–1651, 2014.

S. Wright and J. Nocedal. *Numerical Optimization*, volume 35. 1999.